

**REMARKS**

Claims 1-26 are pending. By this Amendment, Claims 1 and 4 are amended.

Applicant acknowledges the courtesy extended by the Examiner to Applicant's undersigned representative during the October 17, 2007 telephonic interview, the substance of which is incorporated into the comments below.

**Claim Rejections – 35 U.S.C. § 102**

Claims 1-26 were rejected under 35 U.S.C. 102(b) as being anticipated by Hastings (US 5,535,329). This rejection is respectfully traversed.

As stated in MPEP § 2111.01, page 2100-41, *"Where an explicit definition is provided by the applicant for a term, that definition will control interpretation of the term as it is used in the claim."*

Accordingly, the following definitions in numbered paragraphs [0001], [0002] and [0021] of the originally filed application control interpretation of the *"generic function call"* recited in independent Claims 1, 9 and 18:

*"[0001] In common usage today are computer executable functions whose exact mode of operation can vary depending on the nature of the parameters specified by the function. If parameters of one type are input into, and defined for, the function then one operation is performed by the function. Whereas, if a different set of specified parameters are input and defined for the function, then a different operation is performed by the function. Such functions are commonly referred to as generic function calls. A generic function call is a function consisting of a set of methods with a common calling protocol. A generic function computes its value by selecting and calling an appropriate method based on the types of the arguments. Developers can use these generic function calls to conveniently accomplish a vast array of programming tasks."*

*"[0002] Each parameter of a generic function is similarly associated with its own set of distinctive requirements. Each generic function call has associated therewith a parameter list that includes, among other things, all the data characteristics (e.g., types, kinds, ranks, etc.) for each parameter of the generic function call. This list defines a "signature" for the specific function. Although disclosed generically here, the principles of the present invention are intended to be broadly applied to all other programming languages that use generic function calls. Other examples include, without limitation, Fortran generic functions and overloaded function templates in C++ and Java, which are sometimes referred to as ad-hoc polymorphic routines."*

*"[0021] As used herein "generic function calls" are computer executable instructions that are configured to invoke a plurality of different "specific functions" or "methods". The precise operation of the "specific function" depends upon the parameters (also called arguments) of the generic function call. The principles of the present invention can be readily applied to any forms of computer executable instructions that use generic functions. Although not limited to such, examples of such functions can include, but are not limited to overloaded function templates or polymorphic routine dynamic binding in object-oriented languages such as C++ and Java."*

The Examiner argues that the function call recited in Hastings at column 8, line 53 (C8/L53) discloses the claimed generic function call. However, Hastings clarifies at C9/L30-33 that Hastings' function call simply maintains a memory status array. Hastings' function call does not fall within the definitions of "generic function call" explicitly disclosed in the present application, and accordingly Hastings fails to disclose or suggest the "*generic function call*" recited in each of the independent Claims 1, 9 and 18.

Applicant notes further that Hastings is directed to a method of debugging software code, wherein additional code is inserted into code that is to be tested/analyzed. As noted at C8/L50-58, the additional code includes function calls that are inserted before every instruction that includes a memory access, and before some instructions that change the stack pointer, so that (substantially) all of the memory accesses of the program are monitored for purposes of error checking. As noted in C9/L22-56, in particular L31-36, these inserted instructions update a memory

status array with status of the memory and enable checking for errors in writing to, or reading from, memory.

Because Hastings is fundamentally different from the presently claimed invention, the arguments in the Office Action do not make logical sense. Two examples follow.

With respect to the feature of “*first sorting through the relation to determine whether the generic function call contains errors*”, as recited for example in Claim 1, the Examiner cites Hastings at C3/L40-45. However, this citation discloses using a memory status array to check for errors of writing to unallocated memory and reading from unallocated or un-initialized memory. As noted above, the added function calls in Hastings that the Examiner (erroneously) equates with the claimed “*generic function call*” simply update the memory status array. Hastings does not check these added function calls for errors! Instead, Hastings checks the *pre-existing software code* for errors using the added function calls.

In another example, with respect to Claim 3 the Examiner cites C10/L50-64 of Hastings, which describes Hastings’ memory status array. But, the memory status array simply indicates the status of different memory locations or cells, and utterly fails to disclose or suggest “*argument signatures*” that relate sets of “*dummy arguments with associated specific functions for the generic function*”, as recited in Claim 3.

For at least the above reasons, withdrawal of the rejection of Claims 1-26 under 35 U.S.C. § 102(b) over Hastings is respectfully requested.

### CONCLUSION

It is respectfully submitted that each of the presently pending claims is in condition for allowance and notification to that effect is requested. In the event any questions arise regarding this communication or the application in general, the Examiner is invited to contact Applicant's undersigned representative at the telephone number listed below. Although only certain arguments regarding patentability are set forth herein, there may be other arguments and reasons why the claimed invention is patentable. Applicant reserves the right to raise these arguments in the future.

Dated: November 7, 2007

Respectfully submitted,

By   
M. David Ream

Registration No.: 35,333  
DARBY & DARBY P.C.  
P.O. Box 770  
Church Street Station  
New York, New York 10008-0770  
(206) 262-8900 • (212) 527-7701 (Fax)  
Attorneys/Agents For Applicant